Department of Defense

High Level Architecture

Interface Specification

Version 1.2

13 August 1997

TABLE OF CONTENTS

TABLE OF FIGURES

1. General

The High Level Architecture (HLA) Interface Specification is one of the three HLA definition documents. This and the other two HLA definition documents, the HLA Object Model Template and the HLA Rules, along with supporting documents, are in the HLA Technical Library on the DMSO homepage (http://www.dmso.mil).

1.1 Purpose

This document provides a specification for the DoD High Level Architecture (HLA) functional interfaces between federates and the Runtime Infrastructure (RTI). The RTI provides services to federates in a way that is analogous to how a distributed operating system provides services to applications. These interfaces are arranged into the six basic RTI service groups given below:

- Federation Management
- Declaration Management
- Object Management
- Ownership Management
- Time Management
- Data Distribution Management

The six service groups describe the interface between the federates and the RTI, and the software services provided by the RTI for use by HLA federates. The initial set of these services was carefully chosen to be those functions most likely to be required across multiple federations. As a result, federate applications will require most of the services described in this document. The RTI requires a set of services from the federate that are referred to as "RTI Initiated" and are denoted with a †.

1.2 HLA Federation Object Model Framework

A concise and rigorous description of the object model framework is essential to the specification of the interface between federates and the RTI and of the RTI services. The rules and terminology used to describe a federation object model are described in the "High Level Architecture Object Model Template" [HLA OMT]. A Simulation Object Model (SOM) describes salient characteristics of a federate to aid in its reuse and other activities focused on the details of its internal operation and as such is not the concern of the RTI and its services. A Federation Object Model (FOM), on the other hand, deals with inter-federate issues and is relevant to the use of the RTI. The DoD HLA definition states that federation object models describe:

- The set of object classes chosen to represent the real world for a planned federation,

- The set of interaction classes chosen to represent the interplay among real world objects,
- The attribute and parameters of these classes,
- The level of detail at which these classes represent the real world, including all characteristics.

Every object is an instance of an object class found in the FOM. Object classes are chosen by the object model designer to facilitate a desired organizational scheme. Each object class has a set of attributes associated with it. An *attribute* is a distinct, identifiable portion of the object state. In this discussion, "attribute designator " refers to the attribute and "attribute value" refers to its contents. From the federation perspective, the set of all attribute values for a particular object completely defines the state of the object. Federates are free to associate additional state information with an object that is not communicated between federates, but this is outside the HLA federation object model purview.

Federates use the state of the objects as one of the primary means of communication. At any given time only one federate is responsible for simulating a given object attribute. That federate provides new values for that attribute to the other federates in the federation execution through the RTI services. The federate providing the new attribute values is said to be *updating* that attribute value. Federates receiving those values are said to be *reflecting* that attribute.

The privilege to update a value for an attribute is uniquely held by a single federate at any given time during a federation execution. A federate that has the privilege to update values for an attribute is said to *own* that attribute. The RTI provides services that allow federates to exchange ownership of object attributes. The federate that registers an object implicitly owns the privilegeToDeleteObject attribute for that object. The RTI provides services that allow federates to transfer the "privilegeToDeleteObject" attribute in the same way as other attributes.

All objects have an ID. The value of the ID is unique for each federation execution. Object IDs are dynamically generated by an RTI service or can be drawn from a pool of reserved values. These reserved values are set aside for special situations where federates must have knowledge of object IDs before a federation execution begins.

The FOM framework also allows for interaction classes for each object model. The types of interactions possible between different classes of objects, their affected attributes and the interaction parameters are specified. An interaction is an explicit action taken by an object, that can optionally be directed toward another object.

A *federation* is the combination of a particular FOM, a particular set of federates, and the RTI services. A federation is designed for a specific purpose using a commonly understood federation object model and a set of federates that can associate their individual semantics with that object model. A *federation execution* is an instance of the *Create Federation Execution* service invocation and entails executing the federation with a specific FOM, an RTI and using various execution details.

1.3 General Nomenclature and Conventions

There are various entities (classes, attributes, parameters, regions, federates, etc.) referenced in this document which can have the following different "views":

- name - human readable or for communication between federates
- handle - computer manipulable or for communication between a federate and the RTI

The parameters to the services described in this document will use different views of the entities depending on a particular RTI implementation. For clarity, this document refers only to a generic view, known as a "designator", when referring to these entities.

The following sets of data are needed for the implementation of a running RTI and federation executions:

- Federation Execution Data (FED) - information derived from the FOM (class, attribute, parameter names, etc.) and used by the RTI at run-time. Each federation execution needs one. In the abstract, creation of a federation execution is simply the binding of a federation execution name to a FED. The organization of FEDs will become the subject of standardization so Object Model Development Tools can automatically generate them for any vendor's RTI,
- RTI Initialization Data (RID) - RTI vendor specific information needed to run an RTI. A RID is probably supplied when an RTI is initialized.

For all federate initiated services in this specification, except 2.1 Create Federation Execution, 2.2 Destroy Federation Execution, and 2.3 Join Federation Execution, there is an implied supplied parameter which is a federate's connection to a federation execution. For all RTI initiated services there is an implied supplied parameter which is also a federate's connection to a federation execution. The manner in which these parameters are actually provided to the services is RTI implementation dependent, and therefore not shown in the service descriptions.

1.4 Organization Of This Document

The six HLA service groups are specified in sections 2 through 7. Each service is described using several components:

- **Name & Description**

  service name and narrative describing the functionality of the service

- **Service Initiator**

  indicator as to whether the service is RTI or federate initiated

- **Supplied Parameters**

  required and optional service initiator provided parameters

- **Returned Parameters**

  parameters returned by the service

- **Pre-conditions**

conditions which must exist for the service to execute correctly

- **Post-conditions**

  conditions which will exist once the service has executed correctly

- **Exceptions**

  notifications of any irregularity which may occur during service execution

- **Related Services**

  other HLA services which are related to this service

The HLA Application Programmer's Interface (API) is given in section 8 in Common Object Request Broker Architecture (CORBA) Interface Definition Language (IDL) form. section 9 contains the C++ API and section 10 contains the Ada 95 API.

2. Federation Management

Federation Management refers to the creation, dynamic control, modification, and deletion of a federation execution.

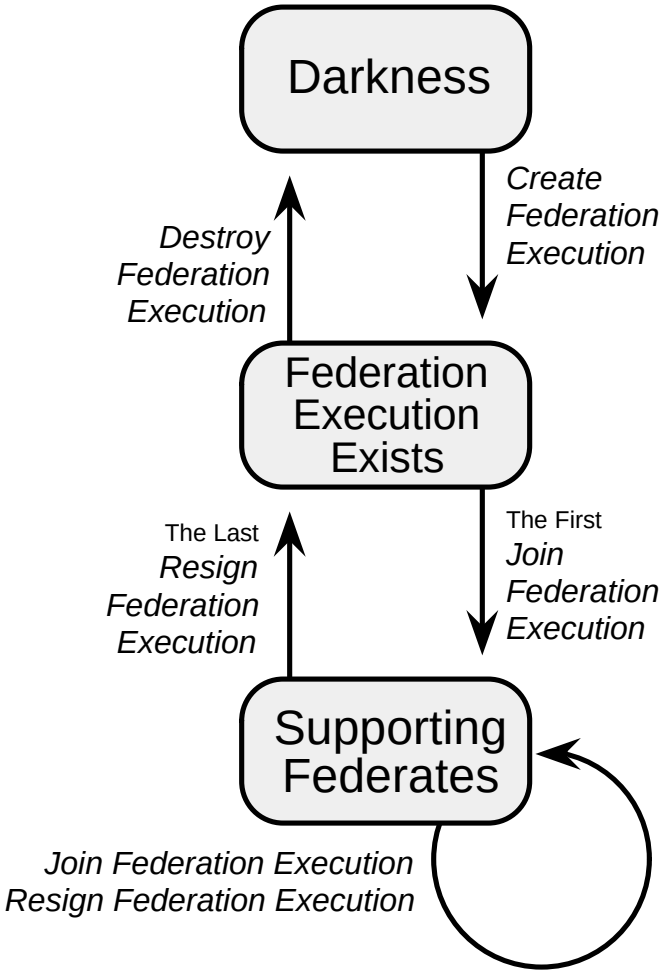

Figure 1 Basic States of the Federation Execution

Before a federate can join a federation execution, the federation execution must exist. Figure 11 shows the overall state of a federation execution as certain basic federation management services are employed.

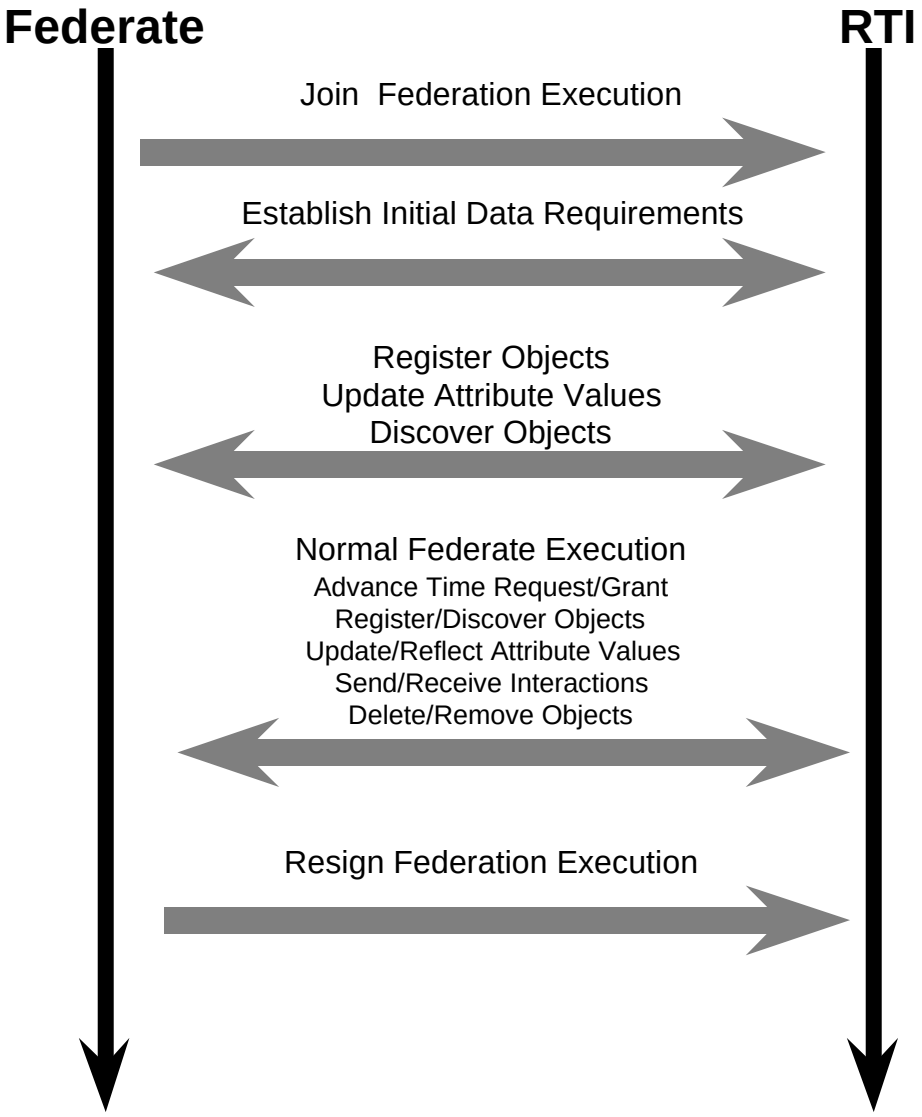

Figure 2 Overall View of Federate to RTI Relationship

Once a federation execution exists, federates can join and resign from it in any sequence that is meaningful to the federation user. Figure 22 presents a generalized view of the basic relationship between a federate and the RTI during the federate participation in a federation execution. The arrows in Figure 22 represent the general invocation of RTI service groups and are not intended to demonstrate strict ordering requirements on the use of the services. The HLA concept does not preclude a single software system from participating in a given federation execution as multiple federates nor does it preclude a given system from participating in multiple (independent) federation executions.

2.1 Create Federation Execution

Federate initiated

The *Create Federation Execution* service creates a new federation execution, and adds it to the set of supported federation executions. Each federation execution created by this service is independent from all other federation executions and there is no inter-communication within the RTI between federation executions. The FED parameter supplies FOM and federation execution data to the RTI.

Supplied Parameters

A federation execution name

A Federation Execution Data designator

- Set of all object classes
- Set of attributes associated with each object class
- Set of interaction classes
- Set of interaction parameters associated with each interaction class
- The ordering and transportation service associated with each attribute of each object class
- The ordering and transportation service associated with each interaction class
- Boolean switch controlling the automatic issue of *Provide Attribute Value Update* service invocations by the RTI on object discovery
- Optionally, routing space designators and the number of dimensions (see section 7 Data Distribution Management)

Returned Parameters

None

Pre-conditions

The federation execution does not exist

Post-conditions

A federation execution exists with the given name that can be joined by federates

Exceptions

The federation execution already exists

Could not locate FED

Invalid FED

RTI internal error

Related Services

Destroy Federation Execution

2.2 Destroy Federation Execution

Federate initiated

This service removes a federation execution from the RTI set of supported federation executions. All federation activity should have stopped and all federates should have resigned before invoking this service.

Supplied Parameters

A federation execution name

Returned Parameters

None

Pre-conditions

There are no federates joined to this federation execution

Post-conditions

The federation execution does not exist

Exceptions

Federates are joined to the federation execution

The federation execution does not exist

RTI internal error

Related Services

Create Federation Execution

2.3 Join Federation Execution

Federate Initiated

The *Join Federation Execution* service affiliates the federate with a federation execution. Execution of the *Join Federation Execution* service indicates the intention to participate in the federation.

Supplied Parameters

A federate designator (denotes the name of the federate attempting to join the federation execution)

A federation execution name

If required, connection parameters that allow the RTI and federate to communicate

Returned Parameters

A joined federate designator (denotes the federate/federation execution connection)

Pre-conditions

The federation execution exists

The federate is not joined to that execution

Post-conditions

The federate is a member of the federation execution

Exceptions

Federate already joined to the federation execution

Specified federation execution does not exist

RTI internal error

Related Services

Resign Federation Execution

2.4 Resign Federation Execution

Federate initiated

The *Resign Federation Execution* service indicates the requested cessation of federation participation. Before resigning, ownership of attributes held by the federate should be resolved. The federate can transfer their ownership to other federates, release them for ownership acquisition at a later time, or delete the object to which they are attached. As a convenience to the federate, the *Resign Federation Execution* service accepts an action parameter that directs the RTI to perform zero, or more, of the following actions:

- delete all objects for which the federate holds that privilege
- release all other attributes for future ownership acquisition - this places the attributes into an unowned state (implying that their values are not being updated), which makes them eligible for ownership by another federate. See section 5 for a more detailed description.

Supplied Parameters

Directive to:

(1) release all attribute ownership

(2) delete all objects for which the federate holds delete privilege

(3) perform action (2) and then action (1)

(4) perform no actions

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The federate is not a member of the federation execution

There are no attributes in the federation execution owned by the federate

Exceptions

Federate owns attributes

Federate not a federation execution member

RTI internal error

Related Services

Join Federation Execution

2.5 Request Pause

Federate initiated

Indicates to the RTI the request to stop the advance of the federation execution. The federation execution members will be instructed by the RTI to pause as soon after the invocation of the *Request Pause* service as possible. The label, supplied when the pause is requested, will be supplied to the other federates via the *Initiate Pause* service.

Supplied Parameters

A label

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federation execution is advancing (not paused)

Post-conditions

A federation pause is pending

Exceptions

Federation already paused

Federate not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Initiate Pause

Pause Achieved

Request Resume

Initiate Resume

Resume Achieved

2.6 Initiate Pause †

RTI Initiated

Instructs the federate to stop changing state as soon as possible. The label provided to the RTI when the pause was requested, via the *Request Pause* service, will be supplied to the federate.

Supplied Parameters

The label supplied when the *Request Pause* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is not already paused

Post-conditions

The federate is informed of a pending pause

Exceptions

Federate cannot pause

Federate already paused

Federate internal error

Related Services

Request Pause

Pause Achieved

2.7 Pause Achieved

Federate Initiated

Indicates that the federate has successfully stopped changing state.

Supplied Parameters

The label supplied when the *Initiate Pause* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is paused

Post-conditions

The RTI is informed that the federate is paused

Exceptions

Unknown label

No pause requested

Federate not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request pause

Initiate Pause

Initiate Resume

2.8 Request Resume

Federate initiated

Indicates the request to resume the advance of the federation execution. The federation execution members will be instructed by the RTI to resume the advance of their state as soon after the invocation of the *Request Resume* service as possible.

Supplied Parameters

None

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federation execution is paused

Post-conditions

A federation resume is pending

Exceptions

Federation not paused

Federate not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Pause

Initiate Resume

Resume Achieved

2.9 Initiate Resume †

RTI Initiated

Informs a paused federate that it may return to the modeling state and start the state evolution in which it was engaged when it invoked *Pause Achieved* t. The federate should resume updating state as soon as possible.

Supplied Parameters

None

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is paused

Post-conditions

The federate is informed that it may resume evolving state

Exceptions

Federate not paused

Federate internal error

Related Services

Request Resume

Resume Achieved

2.10 Resume Achieved

Federate Initiated

Indicates that the federate is evolving state.

Supplied Parameters

None

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is evolving state

Post-conditions

The RTI has been informed that the federate is evolving state

Exceptions

No resume requested

Federate not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Resume

Initiate Resume

2.11 Request Federation Save

Federate initiated

Specifies that a federation save should take place. If the optional federation time parameter is present, the save takes place at that time. If there is no federation time parameter, the federation execution members will be instructed by the RTI to save as soon after the invocation of the *Request Federation Save* service as possible. Only one requested/save will be outstanding at a time. A new save request replaces any outstanding save request.

Supplied Parameters

A label

Optional federation time of the requested federation save

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Save not in progress

Post-conditions

A federation save has been requested

All previous requested saves are canceled

Exceptions

Federation time has already passed (if optional time parameter supplied)

Invalid federation time (if optional time parameter supplied)

Federate not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Initiate Federate Save

Federate Save Begun

Federate Save Achieved

Request Restore

2.12 Initiate Federate Save †

RTI Initiated

Instructs the federate to save state. The federate should save as soon after the invocation of the *Initiate Federate Save* service as possible. The label provided to the RTI when the save was requested, via the *Request Federation Save* service, will be supplied to the federate. A federate can expect a *Initiate Federate Save* invocation anytime it would expect a *Time Advance Grant* invocation.

Supplied Parameters

The label supplied when the *Request Federation Save* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

A federation save has been scheduled

Post-conditions

The federate has been notified to begin saving its state

Exceptions

Unable to perform save

Federate internal error

Related Services

Request Federation Save

Federate Save Begun

Federate Save Achieved

2.13 Federate Save Begun

Federate Initiated

Notifies the RTI that the federate is beginning to save its state.

Supplied Parameters

None

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate has received an *Initiate Federate Save* invocation

The federate is ready to start saving its state

Post-conditions

The RTI has been informed that the federate has begun saving its state

Exceptions

Save not initiated

Federate not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Federation Save

Initiate Federate Save

Federate Save Achieved

2.14 Federate Save Complete

Federate Initiated

Notifies the RTI that the federate has completed its save attempt. The save-success indicator informs the RTI that the federate save either succeeded or failed.

Supplied Parameters

A save-success indicator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate has completed the attempt to save its state

Post-conditions

The RTI has been informed of the status of the state save attempt

Exceptions

Invalid save-success indicator

Save not initiated

Federate not a federation execution member

Restore in progress

RTI internal error

Related Services

Request Federation Save

Initiate Federate Save

Federate Save Begun

2.15 Request Restore

Federate initiated

Directs the RTI to begin the federation execution restoration process.

Supplied Parameters

The label supplied when the *Request Federation Save* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federation has a save with the specified label

Post conditions

Federation restore is pending

Exceptions

No federation save associated with the label

Federate not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Initiate Restore

Restore Achieved

Request Federation Save

2.16 Initiate Restore †

RTI Initiated

Instructs the federate to return to a previously saved state indicated by the supplied federation save label.

Supplied Parameters

The label supplied when the *Request Restore* service was invoked

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate has a save with the specified label

Post-conditions

The federate has been informed to begin restoring state

Exceptions

No federate save associated with the label

Could not initiate restore

Federate internal error

Related Services

Request Restore

Restore Achieved

2.17 Restore Complete

Federate Initiated

Notifies the RTI that the federate has completed its restore attempt. If restore was successful, the federate is in the state it was in when the federation save associated with the label occurred.

Supplied Parameters

The label supplied when the *Initiate Restore* service was invoked

Restore-success indicator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Federate was directed to restore through invocation of the *Initiate Restore* service

If restore was successful, the federate is in a state identical to the state it was in when the federation save associated with the supplied label occurred. If restore was unsuccessful the federate is in an undefined state.

Post-conditions

The RTI has been informed of the status of the restore attempt

Exceptions

Unknown label

Invalid restore-success indicator

Restore not requested

Federate not a federation execution member

Save in progress

RTI internal error

Related Services

Request Restore

Initiate Restore

3. Declaration Management

The HLA declaration management approach requires federates to declare to the RTI their request to both generate and receive object state information. These declarations must be consistent with the Federation Object Model and made using services described in this section. In addition to object state information, the interactions generated and received by a federate must also be declared.

All declaration management services take effect as soon as possible in real-time.

3.1 Publish Object Class

Federate Initiated

The information conveyed by the federate via the *Publish Object Class* service is used in multiple ways. First, it indicates an object class of which the federate may subsequently register instances. Second, it indicates which attributes of the object class the federate is capable of providing to the federation. The federate may do this by registering objects of the class and then updating the attribute values. The federate may also (or alternatively) use ownership management services to acquire attributes of objects registered by another federate (of the same object class) and then update the values of those acquired attributes.

Note that every object has the "privilegeToDeleteObject" attribute that is used to manage the privilege to delete an object. This attribute is like any other attribute and its value may be updated by the owning federate as needed.

Each use of this service replaces all information specified to the RTI in previous service invocations for the same object class and is equivalent to invoking *Unpublish Object Class* (with the specified object class) prior to the subsequent invocation of this service.

Supplied Parameters

An object class designator

Set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The specified object class and attributes are part of the FED

Post-conditions

The federate may now register objects of the specified class and update the specified attributes

Exceptions

Object class not defined in the FED

Attributes not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Unpublish Object Class

Subscribe Object Class Attributes

Register Object

3.2 Unpublish Object Class

Federate Initiated

Informs the RTI that the federate will no longer register object instances of the specified object class. The federate will lose ownership of all owned attributes associated with the specified object class, which means that the federate can no longer update any attribute values of object instances of the specified object class

Supplied Parameters

An object class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The object class is defined in the FED

The federate is publishing the object class

Post-conditions

The federate may not register objects of the specified object class

The federate no longer owns any attributes of instances of the specified object class

Exceptions

Object class not defined in the FED

Federate is not publishing the object class

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Publish Object Class

3.3 Publish Interaction Class

Federate Initiated

Informs the RTI which classes of interactions the federate will send to the federation execution.

Supplied Parameters

An interaction class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is specified in the FED

Post-conditions

The federate may now send interactions of the specified class

Exceptions

Interaction class not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Unpublish Interaction Class

Subscribe Interaction Class

Send Interaction

3.4 Unpublish Interaction Class

Federate Initiated

Informs the RTI that the federate will no longer send interactions of the specified class.

Supplied Parameters

An interaction class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is specified in the FED

The federate is publishing the interaction class

Post-conditions

The federate may not send interactions of the specified interaction class

Exceptions

Interaction class not defined in the FED

Federate not publishing the interaction class

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Publish Interaction Class

3.5 Subscribe Object Class Attributes

Federate Initiated

Specifies an object class for which the RTI is to begin notifying the federate of discovery of instantiated objects when at least one of that object's attributes are in scope. When subscribing to an object class, the federate may also provide a set of attribute designators. The values of only the specified attributes, for all objects discovered as a result of this service invocation, will be provided to the federate from the RTI (via the *Reflect Attribute Values* service). The set of attribute designators may include attributes from the subscribed object class and all super-classes, as defined in the FED.

When an object is discovered, it is not necessarily discovered as being of the same class as the class of which it was registered. The class with which an object is registered is its *registered class*, while the class with which it is discovered by a particular federate is the object's *represented class*. A federate can only discover an object as being of a class to which the federate is subscribed. If an object is registered by another federate as being of a certain class and a federate is subscribed to that object class, when the object is discovered by the subscribing federate it is as the registered class. In this case, the object's registered class and its represented class are one and the same. If an object is registered by another federate as being of a certain class and a federate is not subscribed to that class but is subscribed to a super-class of that class, when the object is discovered by the subscribing federate it is as the subscribed object class (rather than as its registered class). In this case, the object's registered class and its represented class are different. When an object's represented class is a super-class of its registered class, we say that the object has been *promoted* from its registered class to the represented class.

If a federate subscribes to multiple locations in an object class inheritance tree, each relevant object registration can result in at most one object discovery by the subscribing federate. The represented class will be the registered class, if subscribed by the discovering federate. Otherwise, the represented class will be the closest super-class subscribed by the discovering federate.

Each use of this service replaces all information specified to the RTI in any previous service invocations for the same object class.

Invoking this service with an empty set of attributes is equivalent to invoking the *Unsubscribe Object Class* service with the relevant object class.

Supplied Parameters

An object class designator

Set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The object class and attributes are defined in the FED

Post-conditions

The RTI has been informed of the federate's requested subscription

Exceptions

Object class not defined in the FED

Attributes not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Unsubscribe Object Class Attributes

Publish Object Class

Discover Object

Attributes In Scope

Reflect Attribute Values

3.6 Unsubscribe Object Class

Federate Initiated

Informs the RTI that it is to stop notifying the federate of object instance discovery for the specified object class. All in-scope attributes of discovered object instances of the specified object class will go out-of-scope which will subsequently cause the object instances to be removed via *Remove Object* service invocations.

Supplied Parameters

An object class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The object class is defined in the FED

The federate is subscribed to the object class

Post-conditions

The federate will receive no subsequent *Discover Object* service invocations for the specified object class

The federate will receive no subsequent *Update Attribute Values* service invocations for the attributes of the specified object class

Exceptions

Object class not defined in the FED

Federate is not subscribed to the object class

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Subscribe Object Class Attributes

Attributes Out Of Scope †

Remove Object †

3.7 Subscribe Interaction Class

Federate Initiated

Specifies an interaction class for which the RTI should notify the federate of sent interactions.

When an interaction is received by a federate, it is not necessarily of the same class as which it is sent. The class of a sent interaction is its *sent class*, while the class of a received interaction is its *represented class*. A federate can only receive an interaction as being of a class to which the federate is subscribed. If an interaction of a certain class is sent by another federate and a federate is subscribed to that interaction class, when the interaction is received by the subscribing federate it is as the sent class. All parameters included with the sent interaction will be received by the subscribing federate. In this case, the interaction's sent class and its represented class are one and the same. If an interaction of a certain class is sent by another federate and a federate is not subscribed to that class but is subscribed to a super-class of that class, when the iteration is received by the subscribing federate it is as the subscribed interaction class (rather than as its sent class). Only the parameters from the subscribed class and its super-classes will be received. In this case, the interaction's sent class and its represented class as different. When an interaction's represented class is a super-class of its sent class, we say that the interaction has been *promoted* from its sent class to the represented class.

If a federate subscribes to multiple locations in an interaction class inheritance tree, each relevant interaction sent can result in at most one received interaction in the subscribing federate. The represented class will be the sent class, if subscribed by the receiving federate. Otherwise, the represented class will be the closest subscribed super-class.

Supplied Parameters

An interaction class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is defined in the FED

Post-conditions

The RTI will deliver interactions of the specified interaction class to the federate

Exceptions

Interaction class not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Unsubscribe Interaction Class

Publish Interaction Class

Receive Interaction

3.8 Unsubscribe Interaction Class

Federate Initiated

Informs the RTI to no longer notify the federate of sent interactions of the specified interaction class.

Supplied Parameters

An interaction class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is defined in the FED

The federate is subscribed to interaction class

Post-conditions

The RTI will not deliver interactions of the specified interaction class to the federate

Exceptions

Interaction class not defined in the FED

Federate not subscribed to the interaction class

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Subscribe Interaction Class

3.9 Turn Updates On For Object Class †

RTI Initiated

This service notifies the federate that the specified attributes for the specified object class are required somewhere in the federation execution. The federate should commence with the federation agreed upon update scheme for the specified attributes.

Supplied Parameters

An object class

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is publishing the attributes for the object class

Some other federate in the federation execution is subscribed to the attributes of the object class

Post-conditions

The federate has been notified of the requirement for updates of the specified attributes by some other federate in the federation execution

Exceptions

Object class not published

Attribute not published

Federate internal error

Related Services

Turn Updates Off For Object Class †

Publish Object Class

Subscribe Object Class Attributes
Update Attribute Values

3.10 Turn Updates Off For Object Class †

RTI Initiated

This service indicates to the federate that the specified attributes for specified object class are not required anywhere in the federation execution.

Supplied Parameters

An object class

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is publishing the attributes for the object class

No other federate is subscribed to the attributes of the object class

Post-conditions

The federate has been notified that updates of the specified attributes are not required by any other federate in the federation execution

Exceptions

Object class not published

Attribute not published

Federate internal error

Related Services

Turn Updates On For Object Class †

Publish Object Class

Subscribe Object Class Attributes

Update *Attribute* *Values*

Turn Updates On For Object Instance †

RTI Initiated

This service indicates to the federate that the specified attributes for the specified object instance are required somewhere in the federation execution. The federate should commence with the federation agreed upon update scheme for the specified attributes.

Supplied Parameters

Object ID designator

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate owns the attributes

The federate knows about the object instance with the specified ID

Some other federate in the execution is subscribed to the attributes of the object class

Post-conditions

The federate has been notified of the requirement for updates of the specified attributes of the specified object instance by some other federate in the federation execution

Exceptions

Object not known

Attribute not owned

Federate internal error

Related Services

Turn Updates Off For Object Instance †

Publish Object Class

Subscribe Object Class Attributes

Update Attribute Values

3.11 Turn Updates Off For Object Instance †

RTI Initiated

This service indicates to the federate that the specified attributes for the object instance are not required anywhere in the federation execution.

Supplied Parameters

Object ID designator

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate owns the specified attributes

The federate knows about the object instance with the specified ID

No other federate is subscribed to the attributes of the object class

Post-conditions

The federate has been notified that updates of the specified attributes of the specified object instance are not required by any other federate in the federation execution

Exceptions

Object not known

Attribute not owned

Federate internal error

Related Services

Turn Updates On For Object Instance †

Publish Object Class

Subscribe Object Class Attributes

Update Attribute Values

3.12 Turn Interactions On †

RTI Initiated

This service notifies the federate that the specified class of interactions is required somewhere in the federation. The federate should commence with the federation agreed upon scheme for sending interactions of the specified class.

Supplied Parameters

An interaction class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is publishing the interaction class

Some other federate is subscribed to the interaction class

Post-conditions

The federate has been notified that some other federate in the federation execution is subscribed to the interaction class

Exceptions

Interaction class not published

Federate internal error

Related Services

Turn Interactions Off †

Publish Interaction Class

Subscribe Interaction Class

Send Interaction

3.13 Turn Interactions Off †

RTI Initiated

This service indicates to the federate that the specified class of interactions is not required anywhere in the federation..

Supplied Parameters

An interaction class designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is publishing the interaction class

No other federate is subscribed to the interaction class

Post-conditions

The federate has been notified that no other federate in the federation execution is subscribed to the interaction class

Exceptions

Interaction class not published

Federate internal error

Related Services

Turn Interactions On †

Publish Interaction Class

Subscribe Interaction Class

4. Object Management

This group of RTI services deals with the registration, modification, and deletion of objects and the sending and receipt of interactions.

The concept of scope is introduced in this section. The concept of scope is central to the discovery process, as described in section 4.3. An attribute is in-scope to a federate when it is:

subscribed to by the federate,

published by another federate,

part of a registered object instance, and

not owned by the federate.

4.1 Request ID

Federate Initiated

Request federation execution-unique ID numbers. The RTI will not reuse IDs. A set of IDs may be reserved for special purposes and will not be issued by the *Request ID* service.

Supplied Parameters

The requested number of new ID designators

Returned Parameters

Set of ID designators

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The federate has a new set of unique IDs

Exceptions

ID supply exhausted

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

None

4.2 Register Object

Federate Initiated

The RTI creates a unique object ID and links it with an instance of the supplied object class. All attributes of the object class which are currently published by the registering federate are set as owned by the registering federate.

If the optional object instance name parameter is supplied, that name is associated with the object instance.

Supplied Parameters

An object class designator

Optional object instance name

Returned Parameters

Object ID designator

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Object class is defined in the FED

The federate is publishing the object class

If the optional object instance name parameter is supplied, that name is unique

Post-conditions

The returned object ID designator is associated with the object instance

The federate owns the attributes which are currently published for the specified object class

If the optional object instance name parameter is supplied, that name is associated with the object instance

Exceptions

Object class not defined in FED

Federate not publishing the specified object class

Object instance name not unique

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Publish Object Class

Discover Object

4.3 Discover Object †

RTI Initiated

The *Discover Object* service informs the federate to discover an object instance. An object instance is discovered when all of the following occur:

the instance has been registered by another federate,

the instance has not been previously discovered by this federate or has been previously removed, and

at least one attribute of the instance is in-scope for this federate.

Object instances may be discovered for various reasons:

instance was just registered by another federate or

instance was previously removed and an attribute just came into scope.

Supplied Parameters

An object ID designator

An object class designator

Object discovery reason (registered or in-scope)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The object class is published by some federate

The federate is subscribed to the object class

An instance of that class has been registered, with that ID, by another federate

At least one attribute of the object instance in in-scope to the federate

The federate does not know about an object instance with the specified ID

Post-conditions

The object is known to the federate

Exceptions

The federate could not discover the object

Object class not known

Invalid object discovery reason

Federate internal error

Related Services

Register Object

Subscribe Object Class

Attributes In Scope

4.4 Update Attribute Values

Federate Initiated

Provides the current attribute values to the federation for attributes owned by the federate. The federate should supply changed attribute values as specified in the FED. This service, coupled with the *Reflect Attribute Values* service, form the primary data exchange mechanism supported by the RTI. The service returns a federation-unique event retraction designator.

Supplied Parameters

An object ID designator

A set of attribute designator and value pairs

Federation time

A user-supplied tag

Returned Parameters

An event retraction designator

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate owns the attribute for which values are provided

The attributes are defined in the FED

An object instance with the specified ID exists

Post-conditions

The RTI will distribute the new attribute values to subscribing federates

Exceptions

Object not known

Attributes not defined in the FED

The federate does not own the specified attributes

Invalid federation time

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Reflect Attribute Values †

4.5 Reflect Attribute Values †

RTI Initiated

Provides the federate with new values for a discovered attribute. This service, coupled with the *Update Attribute Values* service, forms the primary data exchange mechanism supported by the RTI.

All the attribute/value pairs in an *Update Attribute Values* service invocation for attributes which have identical transportation and message ordering types will be in one corresponding *Reflect Attribute Values* service invocation. An implication of this is that one *Update Attribute Values* invocation could result in multiple *Reflect Attribute Values* invocations in a subscribing federate.

Supplied Parameters

An object ID designator

A set of attribute designator and value pairs

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

The federate is subscribed to the attributes

Federate doesn’t own the attributes

Post-conditions

The new attribute values have been supplied to the federate

Exceptions

Object not known

Attribute not known

Attribute is owned by federate

Invalid federation time

Federate internal error

Related Services

Update Attribute Values

Time Advance Request

Next Event Request

Time Advance Grant

4.6 Send Interaction

Federate Initiated

Sends an interaction into the federation. The interaction parameters may be those in the specified class and all super-classes, as defined in the FED. The service returns a federation-unique event retraction designator.

Supplied Parameters

An interaction class designator

A set of interaction parameter designator and value pairs

Federation time

A user-supplied tag

Returned Parameters

A event retraction designator

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is publishing the interaction class

The interaction class is defined in the FED

The parameters are defined in the FED

Post-conditions

The RTI has received the interaction

Exceptions

Federate not publishing the specified interaction class

Interaction class not defined in FED

Interaction parameter not defined in FED

Invalid federation time

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Time Advance Request

Next Event Request

Time Advance Grant

Receive Interaction

Publish Interaction Class

Retract

4.7 Receive Interaction †

RTI Initiated

Provides the federate with a sent interaction.

Supplied Parameters

An interaction class designator

A set of interaction parameter designator and value pairs

Federation time

A user-supplied tag

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is subscribed to the interaction class

Post-conditions

The federate has received the interaction

Exceptions

Interaction class not known

Interaction parameter not known

Invalid federation time

Federate internal error

Related Services

Retract

Send Interaction

Subscribe Interaction Class

4.8 Delete Object

Federate Initiated

Informs the federation that an object with that ID, owned by the federate, is to be removed from the federation execution. Once the object is removed from the federation execution, its ID cannot be reused. The RTI will use the *Remove Object* service to inform the reflecting federates that the object has been deleted.

Supplied Parameters

An object ID designator

User-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists

The federate has the privilege to delete the object (it owns the privilegeToDeleteObject attribute)

Post-conditions

The federate no longer updates values for the object

The object does not exist in the federation execution

Exceptions

Federate does not own the delete privilege

Object not known

Invalid federation time

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Remove Object

4.9 Remove Object †

RTI Initiated

Informs the federate that an object has been deleted from the federation execution or all attributes of the object are out-of-scope.

Supplied Parameters

An object ID designator

User-supplied tag

Object removal reason (deleted or out-of-scope)

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

Post-conditions

The federate has been advised to remove the object

Exceptions

Object not known

Invalid object removal reason

Federate internal error

Related Services

Delete Object

4.10 Change Attribute Transportation Type

Federate Initiated

The transportation type for each attribute of an object is defaulted from the object class description in the FED. A federate may choose to change the transportation type during execution. Invoking the *Change Attribute Transportation Type* service will change the transportation type for all future *Update Attribute Values* service calls for the specified attributes on the specified object. When the ownership of an attribute is changed, the transportation type reverts to that defined in the FED.

Supplied Parameters

An object ID designator

A set of attribute designators

A transportation type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists

The attributes are defined in the FED

The federate owns the attribute

Post-conditions

The transportation type is changed for the specified attributes

Exceptions

Object not known

Attribute not defined in FED

The federate does not own the specified attributes

Invalid transportation type

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Service

Update Attribute Values

Change Attribute Order Type

4.11 Change Attribute Order Type

Federate Initiated

The data ordering type for each attribute of an object is defaulted from the object class description in the FED. A federate may choose to change the data ordering type during execution. Invoking the *Change Attribute Order Type* service will change the data ordering type for all future *Update Attribute Values* service calls for the specified attributes on the specified object. When the ownership of an attribute is changed, the order type reverts to that defined in the FED.

Supplied Parameters

An object ID designator

A set of attribute designators

A data ordering type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists

The attributes are defined in the FED

The federate owns the attribute

Post-conditions

The data ordering type is changed for the specified attributes

Exceptions

Object not known

Attribute not defined in FED

The federate does not own the specified attributes

Invalid data ordering type

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Service

Update Attribute Values

Change Attribute Transportation Type

4.12 Change Interaction Transportation Type

Federate Initiated

The transportation type for each interaction is defaulted from the interaction class description in the FED. A federate may choose to change the transportation type during execution. Invoking the *Change Interaction Transportation Type* service will change the transportation type for all future *Send Interaction* service calls for the specified interaction class.

Supplied Parameters

An interaction class

A transportation type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is defined in the FED

The federate is publishing the interaction class

Post-conditions

The transportation type is changed for the specified interaction class

Exceptions

Interaction class not defined in FED

Federate not publishing the interaction class

Invalid transportation type

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Service

Send Interaction

Change Interaction Order Type

4.13 Change Interaction Order Type

Federate Initiated

The data ordering type for each interaction is defaulted from the interaction class description in the FED. A federate may choose to change the data ordering type during execution. Invoking the *Change Interaction Order Type* service will change the data ordering type for all future *Send Interaction* service calls for the specified interaction class.

Supplied Parameters

An interaction class

A data ordering type

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is defined in the FED

The federate is publishing the interaction class

Post-conditions

The data ordering type is changed for the specified interaction class

Exceptions

Interaction class not defined in FED

Federate not publishing the interaction class

Invalid data ordering type

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Service

Send Interaction

Change Interaction Transportation Type

4.14 Attributes In Scope †

RTI Initiated

This service notifies the federate that the specified attributes for the object instance are in-scope for the federate. Subsequent to this service invocation, the RTI may issue *Reflect Attribute Values* service invocations for any of the set of attributes for the object instance.

Supplied Parameters

An object ID designator

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

The federate is subscribed to the attributes

The federate does not own the attributes

The attributes are in-scope

Post-conditions

The RTI is allowed to issue *Reflect Attribute Values* service invocations for any of the set of attributes of the object instance

The federate is ready to accept *Reflect Attribute Values* service invocations for any of the set of attributes of the object instance

Exceptions

Object not known

Attribute not known

Federate not subscribed to attribute

Federate owns attribute

Federate internal error

Related Services

Attributes Out Of Scope †

Reflect Attribute Values †

4.15 Attributes Out Of Scope †

RTI Initiated

This service notifies the federate that the specified attributes of the object instance are out-of-scope for the federate. The RTI guarantees not to issue any subsequent *Reflect Attribute Values* service invocations for any of the set of attributes for the object instance.

Supplied Parameters

An object ID designator

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

The federate is subscribed to the attributes

The attributes are out-of-scope

Post-conditions

The RTI guarantees not to issue *Reflect Attribute Values* service invocations for any of the set of attributes of the object instance

Exceptions

Object not known

Attribute not known

Federate not subscribed to attribute

Federate internal error

Related Services

Attributes In Scope †

Reflect Attribute Values †

4.16 Interaction In Scope †

RTI Initiated

This service notifies the federate that the specified interaction is in-scope for the federate. Subsequent to this service invocation, the RTI may issue *Receive Interaction* service invocations for the interaction.

Supplied Parameters

An interaction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is subscribe to the interaction

The interaction is in-scope

Post-conditions

The RTI is allowed to issue *Receive Interaction* service invocations for the interaction

The federate is ready to accept *Receive Interaction* service invocations for the interaction

Exceptions

Interaction not known

Federate is not subscribed to interaction

Federate internal error

Related Services

Interaction Out Of Scope †

Receive Interaction †

4.17 Interaction Out Of Scope †

RTI Initiated

This service notifies the federate that the specified interaction is out-of-scope for the federate. The RTI guarantees not to issue any subsequent *Receive Interaction* service invocations for the interaction.

Supplied Parameters

An interaction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is subscribed to the interaction

The interaction is out-of-scope

Post-conditions

The RTI guarantees not to issue *Receive Interaction* service invocations for the specified interaction

Exceptions

Interaction not known

Federate is not subscribed to interaction

Federate internal error

Related Services

Interaction In Scope †

Receive Interaction †

4.18 Request Attribute Value Update

Federate Initiated

This service is used to stimulate the update of values of specified attributes. When this service is used, the RTI will solicit the current values of the specified attributes from their owners using the *Provide Attribute Value Update* service. When an object class is specified, the RTI will solicit the specified attributes for all the objects of that class. When an object ID is specified, the RTI will solicit the specified attributes for the particular object. The federation time of any resulting *Reflect Attribute Values* service invocations is determined by the updating federate.

Supplied Parameters

Object ID designator or object class designator

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists (when first parameter is an object ID)

The object class and attributes are defined in the FED (when first parameter is an object class)

Post-conditions

The request for the updated attribute values has been received by the RTI

Exceptions

Object not known

Object class not defined in FED

Attributes not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Provide Attribute Value Update †

Update Attribute Values

4.19 Provide Attribute Value Update †

RTI Initiated

Requests the current values for attributes owned by the federate for a given object. The federate should respond to the *Provide Attribute Value Update* service with an invocation of the *Update Attribute Values* service to provide the requested attribute values to the federation.

Supplied Parameters

An object ID designator

A set of attribute designators

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

The federate owns the specified attributes

Post-conditions

The federate has been notified to provide updates of the specified attribute values

Exceptions

Object not known

Attribute not known

Attribute not owned

Federate internal error

Related Services

Request Attribute Value Update

Update Attribute Values

4.20 Retract

Federate Initiated

Event retraction refers to the ability of a federate to retract (sometimes called cancel or unschedule) a previously scheduled event. This is a common discrete-event simulation primitive often used to model interrupts and other preemptive behaviors. Event retraction is also utilized by optimistic federates to implement mechanisms such as "anti-messages." The *Update Attribute Values* and *Send Interaction* RTI services return a designator for the event that is used to specify the event that is to be retracted. See the "HLA Time Management: Design Document" [HLA TM].

Supplied Parameters

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate has previously issued *Update Attribute Values* and *Send Interaction* service calls and obtained the event retraction designators.

Post-conditions

The RTI is informed that the federate requests to retract the specified event

Exceptions

Invalid event retraction designator

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Reflect Retraction

4.21 Reflect Retraction †

RTI Initiated

Event retraction refers to the ability of a federate to retract (sometimes called cancel or unschedule) a previously scheduled event. If the RTI receives a *Retract* service invocation for an event that has already been delivered to a federate, the *Reflect Retraction* service is invoked on the federates that received that event. See the "HLA Time Management: Design Document" [HLA TM].

Supplied Parameters

A event retraction designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The retracted event has been delivered to the federate

Post-conditions

The federate has been directed to retract the specified event

Exceptions

Event not known

Federate internal error

Related Services

Retract

5. Ownership Management

The ownership management group of services allows federates to transfer ownership of object attributes. An attribute is defined as a distinct and identifiable portion of the state of an object and is defined in the FED. Owning an attribute gives a federate the privilege to provide new values to the federation execution for that attribute. Additionally, ownership of the predefined attribute "privilegeToDeleteObject," gives the owning federate the right to remove an object from the federation execution. The RTI will automatically define attribute privilegeToDeleteObject for all object instances. A federate must publish the privilegeToDeleteObject attribute for the object class whose instances it intends to delete. If a federate intends to acquire the privilege to delete objects registered by other federates, it must publish to the privilegeToDeleteObject attribute for those objects' classes. The value of the privilegeToDeleteObject attribute does not affect ownership management services.

In Figure 33, the federate on the left owns an object attribute that it is attempting to transfer to some other federate. In step 1, the federate invokes the *Request Attribute Ownership Divestiture* service with the object ID and designator of the attribute to be transferred and the Negotiated divestiture condition. In step 2, the RTI then invokes the *Request Attribute Ownership Assumption* service on all federates that have indicated the ability to publish this attribute with the *Publish Object Class* service or, if federates were given in the *Request Attribute Ownership Divestiture* invocation, just the specified federates. Subsequent to receipt of an *Request Attribute Ownership Assumption* invocation, a federate issues a *Request Attribute Ownership Acquisition* invocation if it is amenable to owning the attribute, in step 3. The RTI selects a recipient for the attribute ownership from the set of federates that make ownership requests and completes the transfer, in steps 4 and 5, via an *Attribute Ownership Divestiture Notification* to the divesting federate and via an *Attribute Ownership Acquisition Notification* to the acquiring federate.

If no federates "volunteer" to own the attribute, it is up to the federate invoking the *Request Attribute Ownership Divestiture* service to "time out" and perform any appropriate actions.

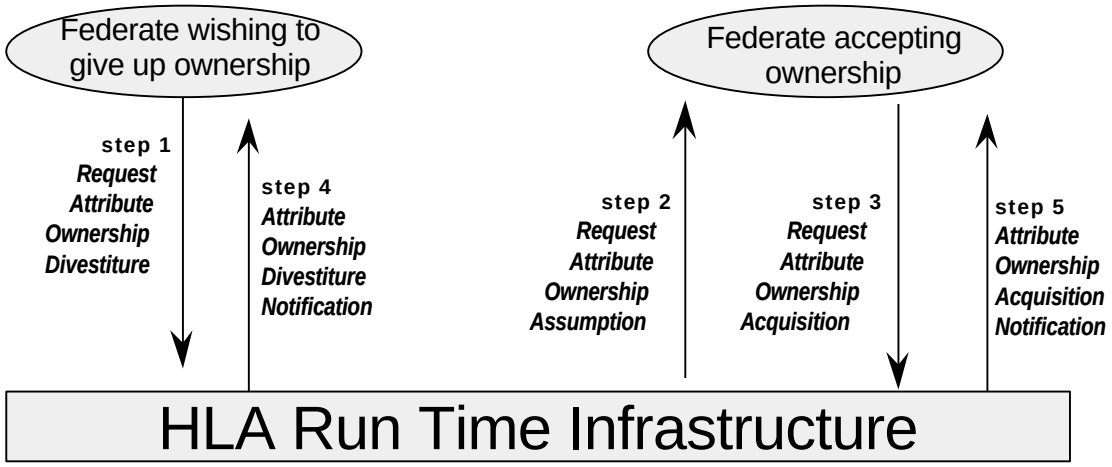

Figure 3 Federate Divesting Attribute Ownership (Negotiated)

If, on the other hand, a federate requests an Unconditional attribute ownership divestiture, the RTI immediately notifies the divesting federate of loss of attribute ownership via the *Attribute Ownership Divestiture Notification* service, the attribute is placed into an unowned state, and the RTI attempts to find an owner. The attribute remains in the unowned state until a) the RTI locates a federate willing to assume ownership or b) a federate requests ownership.

In Figure 44, the federate on the left is attempting to acquire ownership of an object attribute owned by the federate on the right. The federate informs the RTI of this request, in step 1, using the *Request Attribute Ownership Acquisition* service. The RTI locates the current owner of the requested attribute and, in step 2, invokes the *Request Attribute Ownership Release* service providing the attribute owner with the designator of the requested attribute and the acquirer's user supplied tag (perhaps indicating why the transfer is requested). If the attribute owner decides to release ownership, it responds, in step 3, with the designator of the attribute it is releasing via the *Request Attribute Ownership Divestiture* service. Lastly, the RTI completes the transfer by informing the divesting federate it no longer owns the attribute via the *Attribute Ownership Divestiture Notification* service and informing the acquiring federate that it now owns the attribute via the *Attribute Ownership Acquisition Notification* service, steps 4 and 5.

If the owning federate refuses to relinquish ownership of the attribute, it is up to the federate invoking the *Request Attribute Ownership Acquisition* service to "time out" and perform any appropriate actions.

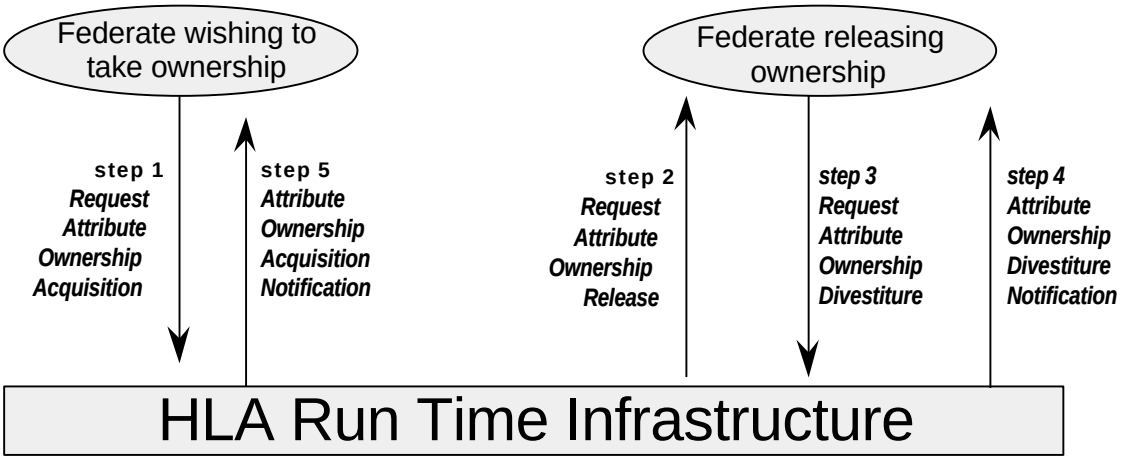

Figure 4 Federate Acquiring Attribute Ownership

5.1 Request Attribute Ownership Divestiture

Federate Initiated

Notifies the RTI that the federate no longer wants to own the specified attributes of the specified object. The federate supplies an object ID and set of attribute designators.

Options:

1. The federate can specify which federate(s) can take ownership of the released attributes, otherwise any federate may own them.
2. The federate can indicate if the requested ownership divestiture is to be negotiated or unconditional. If the divestiture is negotiated, ownership will be transferred only if some federate(s) accepts. An unconditional transfer will relieve the divesting federate of the ownership, causing the attribute(s) to go into (possibly temporarily) the unowned state, without regard to the existence of an accepting federate.

The federate must continue its publication responsibility for the specified attributes until it receives permission to stop via a the *Attribute Ownership Divestiture Notification* service. The federate may receive one or more *Attribute Ownership Divestiture Notification* invocations for each invocation of this service.

Figure 33**Supplied Parameters**

An object ID designator

A set of attribute designators

Ownership divestiture condition (negotiated or unconditional)

A user-supplied tag

Optional set of federates

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists

The federate owns the specified attributes

Post-conditions

No change in attribute ownership

The federate has informed the RTI of its request to divest ownership of the specified attributes

Exceptions

Object not known

Attributes not defined in the FED

Federate does not own attribute

Invalid divestiture condition

Invalid candidate federate

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Attribute Ownership Assumption

Attribute Ownership Divestiture Notification

Attribute Ownership Acquisition Notification

5.2 Request Attribute Ownership Assumption †

RTI Initiated

Informs the federate that the specified attributes for the specified object are available for transfer of ownership to the federate. The RTI supplies an object ID and set of attribute designators. The federate returns the subset of the supplied attribute designators for which it is willing to assume ownership via the *Request Attribute Ownership Acquisition* service. The federate will be notified as to whether or not it received ownership of the attributes in a subsequent invocation of the *Attribute Ownership Acquisition Notification* service.

Figure 33**Supplied Parameters**

An object ID designator

A set of attribute designators

A user-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

The federate is publishing the specified attributes

The federate does not own the specified attributes

Post-conditions

No change in attribute ownership

The federate has been informed of the set of attributes for which the RTI is requesting the federate assume ownership

Exceptions

Object not known

Attribute not known

Federate already owns attribute

Federate not publishing attribute

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Attribute Ownership Divestiture Notification

Attribute Ownership Acquisition Notification

5.3 Attribute Ownership Divestiture Notification †

RTI Initiated

This service notifies the federate that it no longer owns the specified set of attributes. Upon this notification, the federate should stop updating the specified attribute values. The federate may receive multiple notifications for a single invocation of the *Request Attribute Ownership Divestiture* service since different federates may wish to become the owner of different attributes.Figure 33

Supplied Parameters

An object ID designator

The set of released attribute designators

A user-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

The federate owns the specified attributes

A federate has previously attempted to divest ownership of the specified attributes

Post-conditions

The federate does not own the specified attributes

Exceptions

Object not known

Attribute not known

Federate does not own attribute

Federate had not previously attempted to divest ownership of attribute

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Request Attribute Ownership Assumption

Attribute Ownership Acquisition Notification

5.4 Attribute Ownership Acquisition Notification †

RTI Initiated

This service notifies the federate that it now owns the specified set of attributes. The federate may then begin updating those attribute values. The federate may receive multiple notifications for a single invocation of the *Request Attribute Ownership Acquisition* service since the federate may wish to become the owner of attributes owned by different federates.

Figure 33Figure 44**Supplied Parameters**

An object ID designator

A set of acquired attribute designators

A user-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

A federate has previously attempted to acquire ownership of the specified attributes

The specified attributes are not owned by any federate in the federation execution

Post-conditions

The federate owns the specified attributes

Exceptions

Object not known

Attribute not known

Federate had not previously attempted to acquire ownership of the attribute

Federate already owns attribute

Federate internal error

Related Services

Request Attribute Ownership Divestiture

Request Attribute Ownership Assumption

Attribute Ownership Divestiture Notification

5.5 Request Attribute Ownership Acquisition

Federate Initiated

Requests the ownership of the specified attributes of the specified object. The federate supplies an object ID and set of attribute designators. The federate may receive one or more *Attribute Ownership Acquisition Notification* invocations for each invocation of this service.

Supplied Parameters

An object ID designator

A set of attribute designators

A user-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists

The federate is publishing the specified attributes

The federate does not own the specified attributes

Post-conditions

The RTI has been informed of the federates request to acquire ownership of the specified attributes

Exceptions

Object not known

Federate not publishing the object class

Attributes not defined in the FED

Federate not publishing the object attribute

Federate already owns attribute

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Attribute Ownership Release

Attribute Ownership Acquisition Notification

5.6 Request Attribute Ownership Release †

RTI Initiated

Requests that the federate release ownership of the specified attributes of the specified object instance. The *Request Attribute Ownership Release* service provides an object ID and set of attribute designators and is only invoked as the result of a *Request Attribute Ownership Acquisition* service invocation by some other federate. The federate returns the subset of the supplied attributes for which it is willing to release ownership via the *Request Attribute Ownership Divestiture* serviceFigure 44.

Supplied Parameters

An object ID designator

A set of requested attribute designators for release

A user-supplied tag

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

The federate owns the specified attributes

Post-conditions

The federate has been informed of the set of attributes for which the RTI is requesting the federate to release ownership

Exceptions

Object not known

Attribute not known

Federate does not own attribute

Federate internal error

Related Services

Request Attribute Ownership Acquisition

Attribute Ownership Acquisition Notification

5.7 Query Attribute Ownership

Federate initiated

The *Query Attribute Ownership* service is used to determine the owner of the specified attribute of the specified object ID. The federate supplies an object ID and attribute designator. The RTI will provide the attribute owner information via the *Inform Attribute Ownership* service invocation.

Supplied Parameters

An object ID designator

An attribute designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists

Post-conditions

The request for attribute ownership information has been received by the RTI.

Exceptions

Object not known

Attributes not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Inform Attribute Ownership

5.8 Inform Attribute Ownership †

RTI initiated

The *Inform Attribute Ownership* service is used to provide ownership information for the specified attribute of the specified object. This service is invoked by the RTI in response to a *Query Attribute Ownership* service invocation by a federate. This service provides the federate designator of the attribute owner (if the attribute is owned) or an indication that the attribute is available for acquisition.

Supplied Parameters

An object ID designator

An attribute designator

A federate designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate knows about the object instance with the specified ID

Post-conditions

The federate has been informed of the attribute ownership

Exceptions

Object not known

Attribute not known

Federate internal error

Related Services

Query Attribute Ownership

5.9 Is Attribute Owned By Federate

Federate initiated

The *Is Attribute Owned By Federate* service is used to determine if the specified attribute of the specified object ID is owned by the invoking federate. The federate supplies an object ID and attribute designator. The service returns a Boolean value indicating ownership status.

Supplied Parameters

An object ID designator

An attribute designator

Returned Parameters

Attribute ownership indicator

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists

Post-conditions

The federate has the requested ownership information.

Exceptions

Object not known

Attribute not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

None

6. Time Management

Time in the system being modeled is represented in the federation as points along a federation time axis. Each federate advances along this axis during the course of the execution. Such federate time advances may be paced with wallclock time, e.g., as in a training exercise with live participants, or they may not, as in an as-fast-as-possible execution of an analytic model.

Time management is concerned with the mechanisms for controlling the advancement of each federate along the federation time axis. In general, time advances must be coordinated with object management services so that information is delivered to federates in a timely and ordered fashion, e.g., to satisfy requirements for reproducing causal behavior in the system being modeled.

Federates will normally use one of the following approaches to time management:

- *paced, independent time advances.* Federate time advances are *paced* to occur in synchrony with wall clock time (or scaled wall clock time, derived as a linear function of wall clock time). The federate autonomously advances its own time without coordinating such advances with the RTI. "Human-in-the-loop" training federates and "hardware-in-the-loop" test and evaluation federates will typically utilize this approach.
- *paced, coordinated time advances.* Federate time advances are paced to occur in synchrony with (scaled) wall clock time, but time advances are coordinated with the RTI to ensure that before-and-after relationships in the physical system are properly reproduced. Analytic simulation models embedded in environments including humans and/or live elements or physical devices typically utilize this approach.
- *unpaced, coordinated time advances.* Federate time advances are not paced to occur in synchrony with wall clock time, and the federate coordinates time advances to ensure before-and-after relationships are properly modeled. Analytic simulation models intended to execute "as-fast-as-possible" typically utilize this approach.

It is anticipated that some federations may include combinations of federates using different approaches to time management (e.g., coordinated and independent time advance federates may be utilized in a single federation). The time management services are intended to support federations that include federates with different ordering and delivery requirements. For further information, see the "HLA Time Management: Design Document" [HLA TM].

The services described in this section are primarily concerned with coordinated time advance federates. There are three key components of the time management services:

1. *Logical time:* Logical time is synonymous with federation time for coordinated time advance federates. Temporal relationships among events are specified by the logical time values (time stamps) assigned to the events. In general, at any instant during the execution, different coordinated time federates may be at different logical times.
2. *Mechanisms to advance logical time:* Services are provided for each federate to advance its own logical time. Federates must explicitly request logical time advances, and the advance does not occur until the RTI issues a grant. If advances in logical time must be paced to be in synchrony with wallclock time, such pacing must be done within the federate(s).
3. *Message ordering and synchronization:* The RTI will only grant an advance to logical time T when it can guarantee all time stamp ordered (TSO) messages with time stamp less than T (or in some cases less than or equal to T) have been delivered to the federate. This guarantee enables the federate to simulate the behavior of the entities it represents up to logical time T without concern for receiving new events with time stamp less than T.

Each federate using logical time must specify a non-negative *lookahead* value. If a federate's current logical time is T, all TSO messages generated by that federate must have a time stamp value of at least T plus the federate's lookahead. This constraint increases the amount of concurrent execution in the distributed simulation, as described in [HLA TM]. Federates should maximize their lookahead to improve performance.

In order to determine which TSO messages can be delivered to a federate F without violating the guarantee that messages are delivered in time stamp order, the RTI must internally compute a lower bound on the time stamp (LBTS) of future messages that will be later received that are destined for F. Only TSO messages with time stamp less than or equal to F's LBTS value are eligible for delivery. At any instant, the RTI may hold one or more messages with time stamp less than LBTS in its internal queues. The minimum among the federate's LBTS value and messages stored in the RTI's queues gives a lower bound on the time stamp of any TSO messages that will be delivered to the federate in the future.

A federate is said to be *time regulating* (or equivalently, time regulation is enabled) if it can send time stamp ordered (TSO) messages. Information such as the current logical time and lookahead of time regulating federates are used by the RTI in LBTS computations. TSO messages sent by a federate that is *not* time regulating are automatically converted to receive ordered by the RTI. Similarly, a federate is said to be *time constrained* (time constrained is enabled) if it can receive time stamp ordered (TSO) messages. TSO messages received by a federate that is not time constrained are

automatically converted to receive ordered by the RTI.

Enable Time Regulation

Federate Initiated

This service enables time regulation for the federate invoking the service, thereby enabling the federate to send TSO messages. The federate requests that its logical time and lookahead value be set to the values specified as parameters. The RTI may not be able to set the federate's logical time to the value that was requested because doing so might enable the federate to, for example, send a message with a time stamp smaller than the current logical time of another federate. The RTI indicates the logical time assigned to the federate through the *Time Regulating Enabled* service. The logical time that is assigned must be greater than or equal to that requested by the federate.

Supplied Parameters

A value of federation time

A lookahead value

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's request to enable time regulation

Exceptions

Federate is not a federation execution member

Time regulation is already enabled

An *Enable Time Regulation* request is already pending

Save in progress

Restore in progress

RTI internal error

Related Services

Time Regulation Enabled

Disable Time Regulation

Enable Time Constrained

Time Constrained Enabled

Disable Time Constrained

6.2 Time Regulation Enabled †

RTI Initiated

Invocation of this service indicates a prior request to enable time regulation has been honored. The value of the parameter of this service indicates that logical time (LT) of the federate has been set to the specified value.

Supplied Parameters

The current logical time of the federate

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The *Enable Time Regulation* service was previously invoked

Post-conditions

Time regulation is enabled and the federate may now send time stamp ordered (TSO) messages. The federate's logical time is set to the value specified as the parameter to this service. The federate's lookahead is set to that specified in the *Enable Time Regulation* request.

Exceptions

Invalid federation time

Enable Time Regulation was not pending

Federate internal error

Related Services

Enable Time Regulation

Disable Time Regulation

Enable Time Constrained

Time Constrained Enabled

Disable Time Constrained

6.3 Disable Time Regulation

Federate Initiated

Invocation of this service indicates the federate is disabling time regulation. Subsequent Time Stamp Ordered (TSO) messages sent by the federate will be automatically changed to receive ordered.

Supplied Parameters

None

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Time regulation is enabled in the federate

Post-conditions

The federate may no longer send TSO messages

Exceptions

Time Regulation was not enabled

Save in progress

Restore in progress

RTI internal error

Related Services

Enable Time Regulation

Time Regulation Enabled

Enable Time Constrained

Time Constrained Enabled

Disable Time Constrained

6.4 Enable Time Constrained

Federate Initiated

This service requests that the federate invoking the service become time constrained. The RTI indicates the federate is time constrained by invoking the *Time Constrained Enabled* service.

Supplied Parameters

None.

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's request to become time constrained

Exceptions

Federate is not a federation execution member

Time constrained is already enabled

An *Enable Time Constrained* request is already pending

Save in progress

Restore in progress

RTI internal error

Related Services

Enable Time Regulation

Time Regulation Enabled

Disable Time Regulation

Time Constrained Enabled

Disable Time Constrained

6.5 Time Constrained Enabled †

RTI Initiated

Invocation of this service indicates a prior request to become time constrained has been honored. The value of the parameter of this service indicates the current logical time (LT) of the federate.

TSO messages stored in the RTI's internal queues when time constrained becomes enabled that have a time stamp greater than or equal to the federate's logical time will be delivered in time stamp order. TSO messages delivered to the federate prior to it becoming time constrained, possibly including messages with time stamp greater than or equal to the federate's current logical time, will be delivered as receive ordered messages.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The *Enable Time Constrained* service was previously invoked

Post-conditions

The federate may now receive time stamp ordered (TSO) messages and its logical time (LT) advances are constrained so that the federate's LT never exceeds the LBTS value computed by the RTI for the federate. The federate's logical time is set to the value specified as the parameter to this service.

Exceptions

Invalid federation time

Enable Time Constrained was not pending

Federate internal error

Related Services

Enable Time Regulation

Time Regulation Enabled

Disable Time Regulation

Enable Time Constrained

Disable Time Constrained

6.6 Disable Time Constrained

Federate Initiated

Invocation of this service indicates the federate is no longer time constrained. Subsequent Time Stamp Ordered (TSO) messages delivered to the federate will be automatically changed to receive ordered.

Supplied Parameters

None

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is time constrained

Post-conditions

The federate is no longer time constrained, and can no longer receive TSO messages

Exceptions

Time Constrained was not enabled

Save in progress

Restore in progress

RTI internal error

Related Services

Enable Time Regulation

Time Regulation Enabled

Disable Time Regulation

Enable Time Constrained

Time Constrained Enabled

6.7 Request Federation Time

Federate Initiated

Requests a current estimate of the federation time. Federation time is defined to be the minimum of Lower Bound Time Stamp (LBTS) and the current value of the federate's Logical Time (LT).

Supplied Parameters

None

Returned Parameters

The current minimum of LBTS and LT

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The federate receives the current value of federation time

Exceptions

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request LBTS

Request Federate Time

Request Minimum Next Event Time

6.8 Request LBTS

Federate Initiated

Requests the current value of LBTS.

Supplied Parameters

None

Returned Parameters

The current value of LBTS

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The federate receives the current value of LBTS

Exceptions

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Federation Time

Request Federate Time

Request Minimum Next Event Time

6.9 Request Federate Time

Federate Initiated

Requests the current value of the federate's Logical Time (LT).

Supplied Parameters

None

Returned Parameters

The current value of LT.

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The federate receives the current value of LT

Exceptions

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Federation Time

Request LBTS

Request Minimum Next Event Time

6.10 Request Minimum Next Event Time

Federate Initiated

Requests the minimum of LBTS and the time stamp of the next Time Stamp Ordered (TSO) message that is held by the RTI for delivery to the requesting federate, if there are any.

Supplied Parameters

None

Returned Parameters

The minimum of the following:

LBTS and

the time of the message at the head of the TSO queue (if it contains a message).

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The federate receives the minimum next event time.

Exceptions

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Federation Time

Request LBTS

Request Federate Time

6.11 Change Lookahead

Federate Initiated

Change the requested value of the federate's lookahead. The specified lookahead value must be greater than or equal to zero. If the federate is increasing its lookahead, the change takes effect immediately. If the federate is decreasing its lookahead, the decrease takes effect gradually as the federate advances its logical time. Specifically, the federate's lookahead decreases by T units each time logical time advances T units until the requested value is reached.

Supplied Parameters

A requested value of lookahead

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's new value of lookahead

Exceptions

Invalid Lookahead Time

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Request Lookahead

6.12 Request Lookahead

Federate Initiated

Queries the RTI for the current value of the federate's lookahead. The current value of lookahead may differ temporarily from the requested lookahead given in the *Set Lookahead* service if the value of lookahead has been reduced.

Supplied Parameters

None

Returned Parameters

The federate's current value of lookahead

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The federate receives the current value of it's lookahead

Exceptions

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Set Lookahead

6.13 Time Advance Request

Federate Initiated

The *Time Advance Request* service requests an advance of the federate's logical time, and releases zero or more messages for delivery to the federate. Invocation of this service causes the following messages to be delivered to the federate: (i) all incoming receive ordered messages held in the RTI's internal queue, and (ii) all Time Stamp Ordered (TSO) messages with time stamp less than or equal to the specified time. After invoking *Time Advance Request*, the messages are passed to the federate by the RTI invoking the *Receive Interaction* and *Reflect Attribute Values* services. By invoking *Time Advance Request* with the specified time, the federate is guaranteeing that it will not generate a Time Stamp Ordered (TSO) message at any time in the future with time stamp less than or equal to the specified time, even if the federate's lookahead is zero. Further, the federate may not generate any TSO messages in the future with time stamp less than the specified time plus that federate's current lookahead. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the specified time, and no additional TSO messages will be delivered to the federate in the future with time stamp less than or equal to the time of the grant. *Time Advance Request* (or *Time Advance Request Available*) will typically be used by federates using a time-stepped mechanism for advancing logical time.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's request to advance time

Exceptions

Invalid federation time

Time Advance Request, Time Advance Request Available, Next Event Request, Next Event Request Available,, or *Flush Queue Request* already pending

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Time Advance Request Available

Next Event Request

Next Event Request Available

Flush Queue Request

Time Advance Grant

6.14 Time Advance Request Available

Federate Initiated

The *Time Advance Request Available* service requests an advance of the federate's logical time. It is similar to *Time Advance Request* to time T except (1) the RTI does not guarantee delivery of all messages with time stamp equal to T when a Time Advance Grant to time T is issued, and (2) after the federate receives a Time Advance Grant to time T, it can send additional messages with time stamp equal to T if the federate's lookahead value is zero. Invocation of this service causes the following messages to be delivered to the federate: (i) all incoming receive ordered messages held in the RTI's internal queue, (ii) all Time Stamp Ordered (TSO) messages with time stamp less than the specified time, and (iii) any incoming TSO messages held in the RTI's internal queue with time stamp equal to T. After invoking *Time Advance Request Available*, the messages are passed to the federate by the RTI invoking the *Receive Interaction* and *Reflect Attribute Values* services. By invoking *Time Advance Request Available* with the specified time, the federate is guaranteeing that it will not generate a Time Stamp Ordered (TSO) message at any time in the future with time stamp less than the specified time, plus that federate's current lookahead. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the specified time, and no additional TSO messages will be delivered to the federate in the future with time stamp less than the time of the grant. Additional messages with time stamp equal to the time of the grant could later arrive in the future.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's request to advance time

Exceptions

Invalid federation time

Time Advance Request, Time Advance Request Available, Next Event Request, Next Event Request Available,, or *Flush Queue Request* already pending

Federation time already passed

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Time Advance Request

Next Event Request

Next Event Request Available

Flush Queue Request

Time Advance Grant

6.15 Next Event Request

Federate Initiated

The *Next Event Request* service requests the logical time of the federate to be advanced to the time stamp of the next TSO message that will be delivered to the federate, provided that message has a time stamp no greater than the logical time specified in the request. The invocation of this service causes the following messages to be delivered to the federate:

- all receive ordered messages held in the RTI's internal queue, and
- the smallest time stamped TSO message that will ever be delivered in the future with time stamp less than or equal to the specified time, and all other TSO messages containing this same time stamp value that will be generated during the federation execution.

These messages are delivered to the federate by the RTI calling the *Receive Interaction* or *Reflect Attribute Values* services provided by the federate. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the time stamp of the TSO messages that are delivered, if any, or to the specified time if no TSO messages were delivered.

By invoking *Next Event Request* with the specified time the federate is conditionally guaranteeing that it will not generate a Time Stamp Ordered (TSO) message at any time in the future with time stamp less than or equal to the specified time (or less than the specified time plus the federate's lookahead if its lookahead is not zero) if it does not receive any TSO messages as a result of invoking this service. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the time of the grant, and no additional TSO messages will be delivered to the federate in the future with time stamp less than or equal to the time of the grant. *Next Event Request* (or *Next Event Request Available*) will typically be used by event driven federates using the time stamp of the next local event within the federate as the time value for the *Next Event Request*.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's request to advance time

Exceptions

Invalid federation time

Time Advance Request, *Time Advance Request Available*, *Next Event Request*, *Next Event Request Available*, or *Flush Queue Request* already pending

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Time Advance Request

Time Advance Request Available

Next Event Request Available

Flush Queue Request

Time Advance Grant

6.16 Next Event Request Available

Federate Initiated

The *Next Event Request Available* service requests the logical time of the federate to be advanced to the time stamp of the next TSO message that will be delivered to the federate, provided that message has a time stamp no greater than the logical time specified in the request. It is similar to *Next Event Request* except (1) the RTI does not guarantee delivery of all messages with time stamp equal to T when a *Time Advance Grant* to time T is issued, and (2) after the federate receives a *Time Advance Grant* to time T, it can send additional messages with time stamp equal to T if the federate's lookahead value is zero. The invocation of this service causes the following messages to be delivered to the federate:

- all receive ordered messages held in the RTI's internal queue,
- the smallest time stamped TSO message that will ever be delivered in the future with time stamp less than or equal to the specified time, and any other TSO messages containing this same time stamp value that are currently buffered in the RTI's internal queue.

These messages are delivered to the federate by the RTI calling the *Receive Interaction* or *Reflect Attribute Values* services provided by the federate. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the time stamp of the TSO messages that are delivered, if any, or to the specified time if no TSO messages were delivered.

By invoking *Next Event Request Available* with the specified time the federate is conditionally guaranteeing that it will not generate a Time Stamp Ordered (TSO) message at any time in the future with time stamp less than the specified time plus the federate's lookahead if it does not receive any TSO messages as a result of invoking this service. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the time of the grant, and no additional TSO messages will be delivered to the federate in the future with time stamp less than or equal to the time of the grant.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's request to advance time

Exceptions

Invalid federation time

Time Advance Request, Time Advance Request Available, Next Event Request, Next Event Request Available,, or *Flush Queue Request* already pending

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Time Advance Request

Time Advance Request Available

Next Event Request

Flush Queue Request

Time Advance Grant

6.17 Flush Queue Request

Federate Initiated

The *Flush Queue Request* service is similar to *Next Event* Request except all TSO messages stored in the RTI's internal queues are delivered to the federate invoking this service. Time Stamp Ordered (TSO) messages are delivered, despite the fact that the RTI may not be able to guarantee future messages containing a smaller time stamp could arrive. If the federate will not receive any additional TSO messages with time stamp less than the specified time, then the federate's LT will be advanced up to the specified time. Received messages are delivered to the federate by the RTI calling the *Receive Interaction* or *Reflect Attribute Values* services provided by the federate. A *Time Advance Grant* completes this request and indicates to the federate that it has advanced its logical time to the minimum of the specified time, LBTS, and the time stamp of the next TSO message to be received.

Like *Next Event Request*, if a federate invokes *Flush Queue Request* with a time parameter T, the federate conditionally guarantees that it will not schedule any new events with time stamp less than T plus the federate's lookahead if no TSO messages are passed to the federate with time stamp less than T.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The RTI is informed of the federate's request to advance time

Exceptions

Invalid federation time

Time Advance Request, *Time Advance Request Available*, *Next Event Request*, *Next Event Request Available*, or *Flush Queue Request* already pending

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Time Advance Request

Next Event Request

Time Advance Grant

6.18 Time Advance Grant †

RTI Initiated

Invocation of this service indicates a prior request to advance Logical Time (LT) has been honored. The parameter of this service indicates that LT for the federate has been advanced to this value. If the grant is issued in response to invocation of *Next Event Request, Time Advance Request, of Flush Queue Request,* the RTI guarantees no additional Time Stamped Ordered (TSO) messages will be delivered in the future with a time stamp less than or equal to this value. If the grant is in response to an invocation of Time Advance Request Available or Next Event Request Available, the RTI guarantees no additional TSO messages will be delivered in the future with time stamp less than the value of the grant.

Supplied Parameters

A value of federation time

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Any of the *Time Advance Request, Time Advance Request Available, Next Event Request, Next Event Request Available,* or *Flush Queue Request* services were previously invoked

Post-conditions

No additional TSO messages will be delivered with a time stamp less than or equal to the provided time if *Time Advance Request, Next Event Request,* or *Flush Queue Request* had been invoked, or with time stamp less than the provided time if *Time Advance Request Available* or *Next Event Request Available* had been invoked.

Exceptions

Invalid federation time

Time Advance Request, Time Advance Request Available, Next Event Request, Next Event Request Available, or *Flush Queue Request* was not pending

Federate internal error

Related Services

Time Advance Request

Time Advance Request Available

Next Event Request

Next Event Request Available

Flush Queue Request

7. Data Distribution Management

To support efficient data distribution across a federation, the RTI provides a set of services which facilitate the explicit management of data distribution. The fundamental concept to support data distribution is called *routing spaces*. A routing *space* is a multidimensional coordinate system in which federates express an interest for either receiving data or sending data.

To use routing spaces, each federation defines the allowable routing spaces for the federation execution, including the dimensions (variables) of the routing space. Routing spaces are then initialized in the FED with a name and the number of dimensions. Each attribute of every class is assigned to one and only one routing space in the FED. Every interaction class is also assigned to one routing space in the FED.

Routing spaces are used by federates to specify the distribution conditions for the specific data they are sending or expect to receive. Each federate decides which portions (subsets) of those routing spaces are useful to it and specifies (from the federate's perspective) *regions*, or logical areas of interest particular to the federate, by putting bounds (*extents*) on the dimensions of the selected routing space (see Figure 55). The federate then uses these regions to specify conditions under which it:

expects to reflect attribute values and receive interactions and

will provide updates of attribute values and send interactions.

A region is specified by a set of extents which in turn are specified by a set of ranges (bounds) in the dimensions of the routing space. The specification of a single extent is a complete set of ranges, one for each dimension of the routing space. The result is an n-dimensional rectangle, or hyperrectangle, in the routing space. An extent set contains one or more of these hyperrectangles. As a result, a federate can approximate a complex area for a region by specifying multiple, small extents. However, federation designers should balance the utility of such precise regions against the cost in overhead for the DDM services to maintain and manipulate multiple, small extents.

Additionally, Data Distribution Management adds another criterion to the definition of in-scope. When Data Distribution Management is taken into account, an attribute is in-scope to a federate when it is:

subscribed to by the federate,
published by another federate,
part of a registered object instance,
not owned by the federate, and
the relevant subscription and update regions overlap.

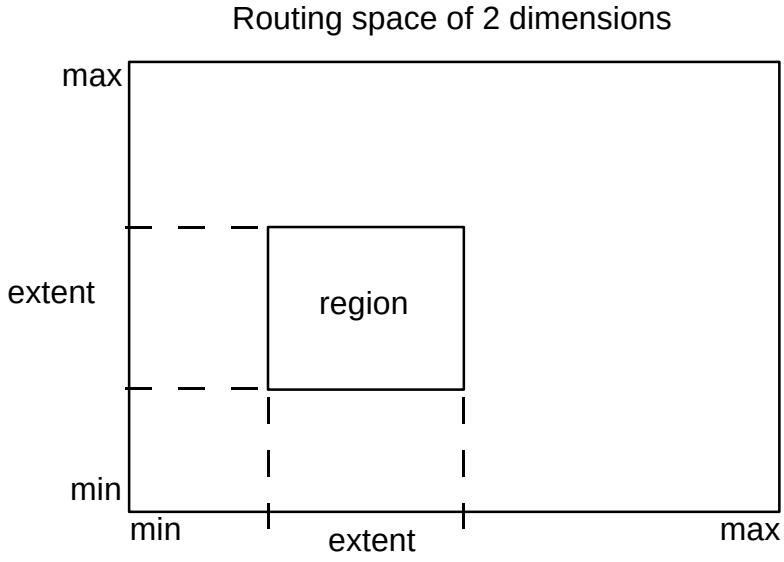

Figure 5 A region and the extents defining it

Specifying a subscription region, the federate notifies the RTI to only deliver data which fall within the extents of the region specified by that federate. Specifying a region and associating that update region with a particular object instance, means that the federate will ensure that the characteristics of the object instance or interaction which map to the dimensions of the routing space fall within the extents of the associated region at the time that the attribute update or send interaction invocation is issued. This implies that the federate is monitoring these added characteristics for each of the attributes owned by the federate. As the state of the objects change, the federate may need to either adjust the extents on the associated regions (*Modify Region*) or change the association to another region (*Associate Region For Updates)*. A set of thresholds is provided to the federate (*Change Thresholds*) indicating how much the extents of a region may change in value before the federate must inform the RTI either by adjusting extents or changing the association.

The routing space, regions, and association information is used by the RTI to distribute data. When update and subscription regions of different federates overlap, the RTI ensures that the attribute updates and interactions associated with that update region are routed to federates with subscription regions which overlap the sender's update region.

It is important to note that the dimensions of routing spaces need not necessarily map to attributes in the FOM. The data distribution management services provide a federation the

capability to specify data distribution on characteristics of objects other than those exchanged as part of federation execution.

Each federate can create multiple update and subscription regions. Update regions may be associated with individual attributes of object instances that have been registered with the RTI. As an example, a federate might have a update region for each sensor system being simulated.

Figure 66 shows one update region (U1) and two subscription regions (S1, S2) within a two dimensional routing space. In this example, U1 and S1 overlap and therefore attributes and interactions associated with U1 will be routed by the RTI to the federate that created S1. In contrast U1 and S2 do not overlap and attributes and interactions will not be routed from U1 to the federate that created S2.

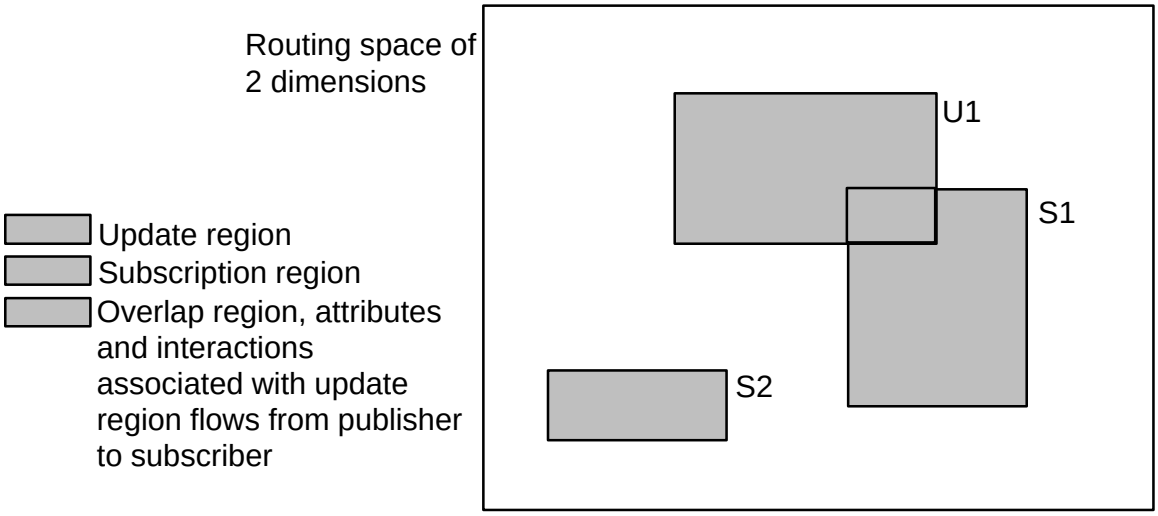

Figure 6 Routing Space Example

Many of the RTI services described in this chapter parallel services defined in sections 2 or 3. Interoperability in federations concurrently employing services from this chapter and chapters 2 or 3 as well as interplay between such services can be understood through the concept of *default regions*. Notionally, a default region covering the full extents of a routing space is implicitly created by the RTI for each routing space specified in the FED. For example, a default region is used for subscriptions using the *Subscribe Object Class Attributes* service. Similarly, attributes are associated with a default region for objects registered via the *Register Object* service. Conceptually, default regions enter into overlap calculations in the same fashion as explicitly specified regions, as described above.

Most data distribution management services take effect as soon as possible in real-time. The exception is *Send Interaction With Region* which has a federation time parameter.

7.1 Create Region

Federate Initiated

Creates a region that is a subset of the specified routing space. The extent set delineates the region within the routing space. The region may be used for either update or subscription.

Supplied Parameters

A routing space designator

A set of extents

Returned Parameters

A region designator

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The routing space is defined in the FED

Post-conditions

A region exists that is a subset of the specified routing space

Exceptions

Routing space not defined in the FED

Invalid extents

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Register Object With Region

Associate Region For Updates

Subscribe Object Class Attributes With Region

Subscribe Interaction Class With Region

Send Interaction With Region

Modify Region

Delete Region

7.2 Modify Region

Federate Initiated

Modify Region changes the delineation of the region to reflect the specified set of extents.

Supplied Parameters

A region designator

A set of extents

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The region exists

Post Conditions

The region has a new extent

Exceptions

Region not known

Invalid extents

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Create Region

Delete Region

Federate Initiated

Deletes the specified region. A region in use for subscription or update cannot be deleted.

Supplied Parameters

A region designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The region exists

The region is not in use

Post-conditions

The region no longer exists

Exceptions

Region not known

Region is use

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Create Region

7.3 Register Object With Region

Federate Initiated

The RTI creates a unique object ID and links it with an instance of the supplied object class. All attributes of the object class which are currently published by the registering federate are set as owned by the registering federate.

This service is used to create an object and simultaneously associate update regions of attributes of that object. This service is an atomic operation that can be used in place of *Register Object* followed by *Associate Region For Updates*. Currently published attributes, for the specified class, which are not supplied in the service invocation are associated with the default region in their respective assigned routing space.

If the optional object instance name parameter is supplied, that name is associated with the object instance.

Supplied Parameters

Object class designator

Set of attribute designator/region designator pairs

Optional object instance name

Returned Parameters

Object ID designator

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

Object class is defined in the FED

The federate is publishing the object class

Attributes are defined in the FED

The federate is publishing the attributes

The regions exist

For each attribute/region pair, the routing space denoted by the region is the same routing space defined for the attribute in the FED

If the optional object instance name parameter is supplied, that name is unique

Post-conditions

The returned object ID designator is associated with the object instance

The federate owns the attributes which are currently published for the specified object class

The regions are associated with the respective attributes of the object instance for future *Update Attribute Values* service invocations

If the optional object instance name parameter is supplied, that name is associated with the object instance

Exceptions

Object class not defined in FED

Federate not publishing the specified object class

Attribute not defined in FED

Federate not publishing the specified attributes

Regions not known

Routing space denoted by region is not the one defined for the object class/attribute in the FED

Object instance name is not unique

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Publish Object Class

Register Object

Create Region

Discover Object †

7.4 Associate Region For Updates

Federate Initiated

This service associates a region to be used for updates with attributes of a specific object instance.

Associating a region with a particular object instance means that the federate will ensure that the characteristics of the object instance which maps to the dimensions of the routing space fall within the extents of the associated region at the time that the attribute update invocation is issued.

The association is used by the *Update Attribute Values* service to route data to subscribers whose subscription regions overlap the specified update region. Based on the object instance and the specified region, this service performs :

- an addition to the group of associations if the object instance/region pair had no attribute set linked with it
- a replacement in the group of associations if there in an attribute set currently linked with the object instance/region pair

Use *Unassociate Region For Updates* service to remove an established association from the group of associations.

Supplied Parameters

An object ID designator

Set of attribute designator/region designator pairs

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The regions exists

The attributes are defined in the FED

For each attribute/region pair, the routing space denoted by the region is the same routing space defined for the object class/attribute in the FED

An object instance with the specified ID exists

The federate owns the specified attributes

Post-conditions

The specified attributes are linked with the respective regions in the association group for future *Update Attribute Values* service invocations

Exceptions

Federate does not own attribute

Region not known

Routing space denoted by region is not the one defined for the object class/attribute in the FED

Object not known

Attributes not defined in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Create Region

Modify Region

Update Attribute Values

Unassociate Region For Updates

7.5 Unassociate Region For Updates

Federate Initiated

This service removes the association between the specified region and all attributes of a specific object instance associated with that region for updates.

Supplied Parameters

An object ID designator

A region designator

Returned Parameters

None

Pre-conditions

The update region is associated with one or more attributes of the object instance

An object instance with the specified ID exists

The federation execution exists

The federate is joined to that federation execution

Post-conditions

The region is unassociated with the specified attributes of the object instance

Exceptions

Region was not associated with the attributes of the object instance

Region not know

Object not known

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Associate Region For Updates

Create Region

Update Attribute Values

Register Object With Region

7.6 Subscribe Object Class Attributes With Region

Federate Initiated

Specifies an object class for which the RTI is to begin notifying the federate of discovery of instantiated objects when at least one of that object's attributes are in scope. This service and subsequent related RTI operations behave similar to *Subscribe Object Class Attributes* and subsequent related RTI operations as described in section 3.5. This service provides additional functionality in that the intersection of the relevant subscription and update regions affect the subsequent RTI operations, as described in the beginning of this section.

There can be only one attribute set linked with each object class/region pair in the group of subscriptions which is defined to support Data Distribution Management. Based on each of the implied object class/region pairs, this service performs one of the following actions with the specified attribute set:

- an addition to the group of subscriptions if the object class/region pair has no attribute set linked with it or
- a replacement in the group of subscriptions if there is currently an attribute set linked with the object class/region pair.

Invoking this service with an set of attributes is equivalent to invoking the *Unsubscribe Object Class With Region* service with the relevant object class.

Supplied Parameters

An object class designator

Set of attribute designator/region designator pairs

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The object class is defined in the FED

The attributes are defined in the FED

The regions exist

For each attribute/region pair, the routing space denoted by the region is the same routing space defined for the object class/attribute in the FED

Post-conditions

The RTI has been informed of the federate's requested subscription

Exceptions

Object class not defined in the FED

Attributes not defined in the FED

Region not known

Routing space denoted by region is not the one defined for the object class/attribute in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Create Region

Unsubscribe Object Class With Region

Publish Object Class

Discover Object †

Attributes In Scope †

Reflect Attribute Values †

7.7 Unsubscribe Object Class With Region

Federate Initiated

Informs the RTI that it is to stop notifying the federate of object instance discovery for the specified object class. The unsubscribe is confined to all subscriptions using the specified region.

Supplied Parameters

An object class designator

A region designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The object class is defined in the FED

The object class is subscribed to the region

The region exist

Post-conditions

The RTI has been informed of the federate's requested unsubscription

Exceptions

Object class not defined in the FED

Region not known

Object class not subscribed to region

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Publish Object Class

Subscribe Object Class Attributes With Region

7.8 Subscribe Interaction Class With Region

Federate Initiated

Specifies the class of interactions which should be delivered to the federate, taking the region into account. This service and subsequent related RTI operations behave similar to *Subscribe Interaction Class* and subsequent related RTI operations as described in 3.7. This service provides additional functionality in that the intersection of the relevant subscription and interaction send regions affect the subsequent RTI operations, as described in the beginning of this section.

If the specified region is currently in the group of regions associated with the specified interaction class subscription, this service performs a replacement in the group, otherwise an addition to the group is performed.

Supplied Parameters

An interaction class designator

A region designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is defined in the FED

The region exists

The routing space denoted by the region is the same routing space defined for the interaction class in the FED

Post-conditions

The RTI will deliver interactions of the specified interaction class sent to the specified region

Exceptions

Interaction class not defined in the FED

Region not known

Routing space denoted by region is not the one defined for the interaction class in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Publish Interaction Class With Region

Unsubscribe Interaction Class with Region

Send Interaction With Region

Create Region

7.9 Unsubscribe Interaction Class With Region

Federate Initiated

Informs the RTI to no longer notify the federate of sent interactions of the specified class which are in the specified region.

Supplied Parameters

An interaction class designator

A region designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The interaction class is defined in the FED

The region exists

The routing space denoted by the region is the same routing space defined for the interaction class in the FED

Post-conditions

The RTI will no longer deliver interactions of the specified interaction class sent to the specified region

Exceptions

Interaction class not defined in the FED

Region not known

Routing space denoted by region is not the one defined for the interaction class in the FED

Federate is not subscribed to the interaction class

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Subscribe Interaction Class with Region

7.10 Send Interaction With Region

Federate Initiated

Sends an interaction into the federation. The interaction parameters may be those in the specified class and all super-classes, as defined in the FED. The region is used to limit the scope of potential receivers of the interaction. The service returns a federation-unique event retraction designator.

Supplied Parameters

An interaction class designator

A set of parameter designator and value pairs

Federation time

A user-supplied tag

A region designator

Returned Parameters

An event retraction designator

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The federate is publishing the interaction class

The interaction class is defined in the FED

The parameters are defined in the FED

The region exists

The routing space denoted by the region is the same routing space defined for the interaction class in the FED

Post-conditions

The RTI has received the interaction

Exceptions

Federate not publishing the specified interaction class

Interaction class not defined in FED

Interaction parameter not defined in FED

Invalid federation time

Region not known

Routing space denoted by region is not the one defined for the interaction class in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Time Advance Request

Next Event Request

Time Advance Grant †

Receive Interaction †

Publish Interaction Class

Retract

Create Region

7.11 Request Attribute Value Update With Region

Federate Initiated

This service is used to stimulate the update of values of specified attributes. When this service is used, the RTI will solicit the current values of the specified attributes from their owners using the *Provide Attribute Value Update* service. When an object class is specified, the RTI will solicit the specified attributes for all the objects of that class. When an object ID is specified, the RTI will solicit the specified attributes for the particular object. The specific requests issued are restricted in accordance with the specified region parameter. The federation time of any resulting *Reflect Attribute Values* service invocations is determined by the updating federate.

Supplied Parameters

Object ID designator or object class designator

A set of attribute designator/region designator pairs

A region designator

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

An object instance with the specified ID exists (when first parameter is an object ID)

The object class is defined in the FED (when first parameter is an object class)

The attributes are defined in the FED

The regions exists

For each attribute/region pair, the routing space denoted by the region is the same routing space defined for the object class/attribute in the FED

Post-conditions

The request for the updated attribute values has been received by the RTI

Exceptions

Object not known

Object class not defined in FED

Attributes not defined in the FED

Region not known

Routing space denoted by the region is not the one defined for the object class/attribute in the FED

Federate is not a federation execution member

Save in progress

Restore in progress

RTI internal error

Related Services

Provide Attribute Value Update †

Update Attribute Values

Create Region

7.12 Change Thresholds †

RTI Initiated

Change Thresholds provides the federate with the new values of the thresholds for each of the dimensions of a routing space.

Supplied Parameters

A region designator

A set of thresholds

Returned Parameters

None

Pre-conditions

The federation execution exists

The federate is joined to that federation execution

The region exists

Post-conditions

The federate is informed of the new threshold values for the region

Exceptions

Region not known

Federate internal error

Related Services

Create Region

Modify Region

8. HLA IDL Application Programmer's Interface

9. HLA C++ Application Programmer's Interface

10. HLA Ada 95 Application Programmer's Interface

11. References

[HLA DEF] "Preliminary Definition, High Level Architecture," briefing, Defense Modeling and Simulation Office, latest version on WEB.

[HLA OMT] "High Level Architecture Object Model Template", Defense Modeling and Simulation Office, latest version on WEB.

[HLA DDM] "HLA Data Distribution Management Design Document", Defense Modeling and Simulation Office, version release TBD.

[HLA TM] "HLA Time Management: Design Document", Defense Modeling and Simulation Office, latest version on WEB.

COMMENTS

Comments on this document should be sent by electronic mail to the Defense Modeling and Simulation Office (DMSO) HLA Specifications e-mail address (hla_specs@msis.dmso.mil). The subject line of the message should include the section number referenced in the comment. The body of each submittal should include: (1) the name and E-mail address of the person making the comment (separate from the mail header), (2) reference to the RTI service or portion of the I/F Specification that the comment addresses (by section and page number), (3) a one sentence summary of the comment/issue, (4) a brief description of the comment/issue, and (5) any suggested resolution or action to be taken.